Graph Rewriting in Topology IV : Rewriting Based on Algebraic Operators (Algorithms in Algebraic Systems and Computation Theory)

AUTHOR(S):

Liu, Jian-Qin; Shimohara, Katsunori

数理解析研究所講究録 1268 巻 2002 年 64-72

Graph Rewriting in Topology IV: Rewriting Based on Algebraic Operators*

Jian-Qin Liu and Katsunori Shimohara

ATR Human Information Science Laboratories,
Hikaridai 2-2-2, Seika-cho, Soraku-gun, Kyoto, 619-0288, Japan

Abstract

Graph rewriting in topology (denoted as GRiT) refers to a kind of rewriting systems on hypergraphs which is expected to be helpful to the study of the parallel computing based on rewriting theory. Under certain topological conditions (here refers to homology and homotopy), we discuss the relation between the formalized GRiT and derived algebraic features from the rewriting operators.

1. Preliminaries

The description of the computation model GRiT is based on the terms of (hyper)graph rewriting[1], bigraphical reactive systems[2] and category theory.
In this section, the main notations [1,2,3,4] used for our discussion are given as follwos:
Let A be an alphabet set, $\tau(a)$ be the rank associated to a symbol ($a \in A$), V_H be a vertex set, E_H be a hyperedge set, $\boldsymbol{H} = \{H\}$, H be a hypergraph, i.e.,

$$H = < V_H, E_H >,$$

where $V_H \cap E_H = \emptyset$.
For more symbols concerned here, let $lab_H(e)$ be the label assigned to a hyperedge e in A. $\tau(lab_H(e))$ be the length of the sequence of vertices with $lab_H(e)$.
Now we introduce a predicate "pah" for the description of pathways to be used later:

$$pah\ (x, y_1, \ldots, y_{L1}, z_1, \ldots, z_{L2})$$

where $L_1, L_2 \in N$,
$x \in E_H$,
$y_1, \ldots, y_{L1} \in V_H$,
$lab_H(x) = a$,
$n = \tau(a)$,
$z_1, \ldots\ldots, z_{L2}$ are the controls in the set of K_H, which is called signature,

* This research was conducted as part of "Research on Human Communication" with funding from the Telecommunications Advancement Organization of Japan.

$$\text{i.e., } K_H = \{z_1, \ldots\ldots, z_{L2}\}.$$

The hyperedge x is the key to the interactions and other coupling relations derived from it, with aspect to the nodes, i.e., the vertexes $y_1, \ldots, y_{L1}$.
Here the controls z, ..., z_{L2} refer to the types of the related bigraphs which includes the vertexes $y_1, \ldots, y_{L1}$ and are assigned with their arity, e.g., arity (y1) = 3, which is just for example that does not mean any specific values are necessary in our discussion.
So we can have the relation

$$R_p(A) = \{pah_a \mid a \in A\},$$

where pah_a is $(\tau(a)+2)$-ary. This relation is defined for the object set of pathways.
Considering the different types of the bigraphs owing to the controls are atomic or active (non-atomic), so, let $PATs_H$ be the equivalent class of pathways corresponding to the signature K_H. Of course, we notice that the equivalent classes will be different for the pathways represented by bigraphs and other bigraphs concerned. But we limit our objects within the domain of the former without explanation in this abstract.
Then we define the structure for pathways corresponding to the relation $R_p(A)$:

$$|H|_3 := < V_H \cup E_H \cup K_H, (pah_{aH})_{a \in A} > \in STR(R_p(A)).$$

The index 3 in $|H|_3$ shows its difference from the $|H|_1$ and $|H|_2$ in [1].
Let $\equiv_H$ be the equivalent relationship of $PATs_H$, the conditions for constructing operators for generating the pathways concerned become one of the most important targets. We select the topological constraints exerted on the vertexes during the rewriting processes as the starting point of observing their features in the framework of rewriting in a three dimensional space.
From another side of the observation, we also try to study the semantics of the formal systems derived from the objects mentioned above. This is the basis of the formalization and further studies on the parallel features of the proposed model GRiT[3,4] such as congruency and operability (i.e., controllability).
Let Tog_H be the category of topographs, Mog_H be the category of monographs, Big_H be the category of bigraphs. We reiterate here that K_H is the control set corresponding to the G_H.
Let Cag_H be the category defined based on the set of K_H, BRS_H be the category of bigraphical reactive systems. Then, we can define the representation for hypergraphs by bigraph forms as

$$G_H = G(U_H, ctrl_H, G^T_H, G^M_H): \iota_m \to \iota_n,$$

where $U_H \subseteq V_H \cup E_H$,
$ctrl_H \in K_H$,
G^T_H in Tog_H,
G^M_H in Mog_H,
G_H in Big_H,
$m, n \in N$.

After the hypergraph has been limited as the subsets of pathways which is the special kinds of hypergraphs, we can select the transduction as the form of rewriting process instead of the rules of rewriting. So, we give the predicate set for the transductions as:

$$\{TRANS(G^{T}_{H}), TRANS(G^{M}_{H})\}$$

Here, $TRANS(G^{T}_{H})$ refers to the predicate set for the transductions exerted on the topographs:

$$G^{T}_{H} = (V, ctrl_{H}, prt): \iota_{m} \rightarrow \iota_{n},$$

where *prt* is the parent of the vertexes in rewriting.

$TRANS(G^{M}_{H})$ refers to the predicate set for the transductions exerted on the monographs:

$$G^{M}_{H} = (V, ctrl_{H}, \equiv_{H}).$$

2. *Formalization*

Besides the engineering practices, parallel computing is also a good place in which we can apply the theoretical computer science. Algebraic theory has offered us a functional way to guide the designing works of parallel algorithms in rigorous theory[1]. Teruo Imaoka points out in the preface of the RIMS Kokyuroku 1222 [5] that "the researches on the structures of algebraic systems in the view of algorithms and computer science are becoming active and a new field called computational algebra is emerging" (his original sentences are written in Japanese). The successful research results achieved by Yuji Kobayashi and Masashi Katsura in [6], Inamu Inata and Yuji Kobayashi in [7] are significant. In this abstract, we focus on the questions how to explore the algebraic operators in the interactions-like processes of parallel computing in the form of (hyper)graph rewriting.

Comparing to the transduction at the abstract level of the abstract machines which functional equals to the rewriting rules based on HR and VR[1] and others[3], we focus on the generating processes of GRiT, i.e., we aim at the interactions of the pathways from the processes with initial simple structures into more complex ones. Let $react_{H}$ be the operators of the interactions of bigraphs based on the transduction of pathways within cag_{H} for the mapping:

$$\iota_{m} \rightarrow \iota_{n} \quad \text{in the } G_{H},$$

the input and output of pathways pah(.) in $R_{p}(A)$. The core of the generating processed lies in the "regulation" mechanism that makes the directions of the "dynamical" operations on the various neighborhoods randomly selected simultaneously. The term "regulation"

---

[1] Masami Ito suggests and encourages our work on hypergraph rewriting by algebraic theory that includes semi-group and codes.

means the mechanism within the programs that can control the modules of the underlying software systems (Cf. Fig.1).

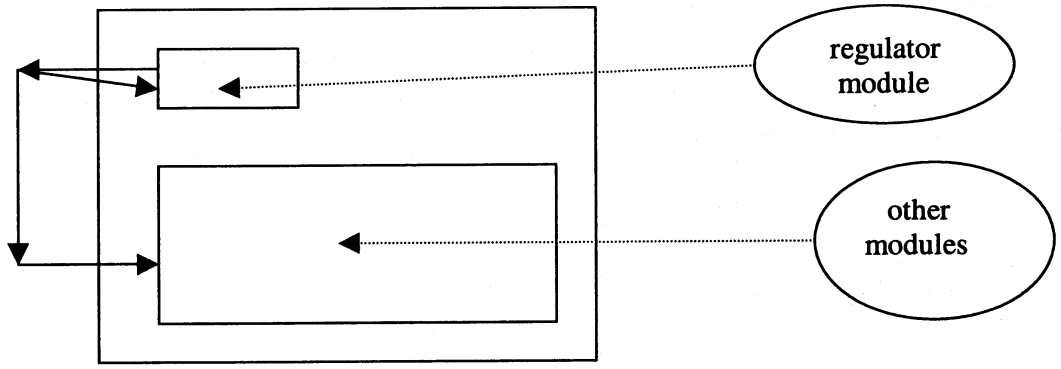

Fig.1 The structure of modules

This mechanism we proposed is enlightened from the kinase-guided bio-chemical reaction processes. In the cells, the kinase can control/regulate the bio-chemical reactions guided by it, which factors include the kinase itself where the self-assembly (self-organizing) play an important role. In the view of biologically inspired information processing systems, the coupling relationship is explored in our scheme for parallel computing without any prerequisite for de-coupling operations.

From the works on the designing the proper operators, we can achieve more useful schemes of efficiently manipulating the interactions leading to developing faster algorithms probably by unconventional paradigms such as molecular computing.
We define

$$cGRiT(|H|_3, Big_H(K_H, react_H))$$

as the rewriting system on the hypergraphs with certain constraints (e.g. topological conditions) which is different from the constraints in[8], constructed by transduction on

$$Big_H(K_H, react_H),$$

where $react_H$ is used in BRS_H. Notice that the formal system we discuss here is functionally equivalent to the "construct" representation in [3] under certain conditions. Based on category cag_H, we can get the operation by predicates on the

$$GRT(|H|_3, Big_H(K_H, react_H), \Im_H)$$

as:

$$\begin{aligned} &\text{INTERACTION } (pah\ (x, y_1, \ldots, y_{L1}, z_1, \ldots, z_{L2}),\\ &\quad pah\ (x', y_1, \ldots, y_{L3}, z_1, \ldots, z_{L4}), \end{aligned}$$

$$GRT(|H|_3, Big_H(K_H, react_H), \Im_H)),$$

s.t. certain topological constraints.

which refers to the fact that x and x' share certain common parts of the hyperedges, $y_1, \ldots, y_{L1}$ in path way one, and $y_1, \ldots, y_{L3}$ in another pathway. Here L_3 and $L_4 \in N$. These two sets also share certain common parts of vertexes, and the neighborhood for the operation exerted is δ_H for the V_H and E_H in

$$GRT(|H|_3, Big_H(K_H, react_H), \Im_H).$$

Within the equivalent class PAT_H of pathways, the transductions are exerted on the topographs:

$$G^T{}_H = (V, ctrl_H, prt): m \to n,$$

where *prt* is the parent of the vertexes in rewriting, $m, n \in N$.
Provided that $ctrl^1{}_H \in K_H$, $ctrl^2{}_H \in K_H$, where $ctrl^1{}_H \neq ctrl^2{}_H$, judging the "capacity of transduction" is important, so we have that

Proposition 1:
There exists that $mono_H$, the set of monoid operators that is inferred from the interactions on $(pah_{aH})_{a\in A}$ of

$$GRiT(|H|_3, Big_H(K_H, react_H)),$$

which satisfies the condition of McNaughton languages for

$$G_H = G(U_H, ctrl_H, G^T{}_H, G^M{}_H): I^* \to J^*.$$

Proposition 2:
The operations of $mono_H$ can keep the congruence of the (hyper)graph rewriting on

$$G_H = G(U_H, ctrl_H, G^T{}_H, G^M{}_H): I^* \to J^*,$$

if they satisfy the condition of McNaughton languages for $I^* \to J^*$ and the interactions of $mono_H$ can be inferred by

$$TRANDN(pah(x, y_1, \ldots, y_m, z_1, \ldots, z_p \ldots z_q, \ldots, z_m))$$
$$\wedge\ VALPATH(pah(x, y_1, \ldots, y_m, z_1, \ldots, z_p \ldots z_q, \ldots, z_k)),$$

where $I^*, J^*, p,q,m,n,k \in N$ for $GRiT(|H|_3, Big_H(K_H, react_H))$.

3. The Algorithm

In GRiT systems, we can use rewriting rule in [3], which is directly operated on the vertex and hyperedges in the VR and HR way. This kind of objects is concrete but the contents of the rewriting rules are abstract with respect to the understandable level of the meanings of programs in details. The transduction form itself is also abstract, but the logic expression is explicit for description of the behavior of the operation and designing schemes for the algorithms derived consequently. It is an effort on parallel algorithms designed by logic guidance if corresponding programming issues are feasible.
Through the proposed algorithm, we are trying to efficiently program and demonstrate the quantitative relationship between feasibility of "self-regulation" and the "complexity" of the underlying mechanism of the computing processes. Notice the term "complexity" is not defined as the complexity in computation theory and mainly refers to the cost-related measurement of the computing process we are discussed.
Let the input of the model be $\{X_i\}$. The pathways are interacted and sustained in recursively computing process. The output of the model is the set of pathways that satisfies the criteria we gave in advance.

The computing process consists of four major steps as follows:

Step 0: Initializing the pathways as atomic forms.

Step1: Interacting the pathways.

New pathways are generated by interactions of the existing pathways. Here, the neighboring pathways are selected as the objects for interactions. The measurement for neighborhood δ_Θ is determined according to the topological constraints. The minimum Hamming distance of the variables of "candidates" is one of the simplest among them.
Let δ_{kl} be the neighborhood of pathway Ψ_k and Ψ_l. The interaction of Ψ_k and Ψ_l in δ_{kl} is made by connecting their pathways to couple the common reactants for "reactions". This means that Ψ_k and Ψ_l are selected in δ_{kl} for k=0,1, ... and l=0, 1, ...
Then we apply transduction on the pathway $\{\Psi_k\}$ (k=0,1, ...) in the neighborhood δ_Θ ($\delta_{kl} \subseteq \delta_\Theta$) and activate the related pathways .
For the quantitative representation, the four main predicates that we define for GRiT are:

(1) VALPATH(pah (x, y_1, ..., y_m, z_1, ..., z_n)):

It is defined as the predicate for validness of pathway, i.e.,

VALPATH(pah (x, y_1, ..., y_m, z_1, ..., z_n))
= true if pah in hGRiT($|H|_3$, Big_H (K_H, $react_H$)),
= false if pah NOT in hGRiT($|H|_3$, Big_H (K_H, $react_H$)).

(2) ELELSN(pah (x, y_1, ..., y_m, z_1, ..., z_n)):

It is defined as the predicate to show the situation of eliminating elision in pathways in hGRiT($|H|_3$, Big_H (K_H, $react_H$)), i.e.,

ELELSN(pah (x, y_1, ..., y_m, z_1, ..., z_n))
= true if pah is active in hGRiT($|H|_3$, Big_H (K_H, $react_H$)),
= false if pah is atomic in hGRiT($|H|_3$, Big_H (K_H, $react_H$)).

The activating process is the generating mechanism of "complexity" increasing with certain probability determined by our parameter setting.

(3) INTERACTION (pah (x, y_1, ..., y_{L1}, z_1, ..., z_{L2}),
pah (x', y_1, ..., y_{L3}, z_1, ..., z_{L4}),
GRT($|H|_3$, Big_H (K_H, $react_H$), $\Im_H$)),
s.t. the topological constraints.

It is the same as we defined in the previous section.

(4) TRANDN(pah (x, y_1, ..., y_m, z_1, ..., z_n):

It is defined as the predicate for transduction, i.e.,

TRANDN(pah (x, y_1, ..., y_m, z_1, ..., z_n))
= true if the sequence {z_1, ..., z_n} can deduce the redexed RPO,
= false else.

For the I* -> J* in GRiT($|H|_3$, Big_H (K_H, $react_H$),$\Im_H$), the truth value of the formula

TRANDN (pah (x, y_1, ..., y_m, z_1, ..., z_n))
$\wedge$ ELELSN (pah (x, y_1, ..., y_m, z_1, ..., z_n))
$\wedge$ VALPATH (pah (x, y_1, ..., y_m, z_1, ..., z_n))

is used to judge the pathways obtained from interactions. Then, the pathway generation is verified by the logic forms.
Let input-node(Ψ_k) be the input "reactant" of pathway Ψ_k, let output-node(Ψ_k) be the output "reactant" of pathway Ψ_k, and let internal-node(Ψ_k) be the internal "reactants" of pathway Ψ_k, where Ψ_k is in the set of all pathways, which implies that pathway $\Psi_{k'}$ covers the reactant molecules that correspond to the sets of sub-pathways. The same parts are kept only once, and these are sustained by pathway Ψ_k. The same is also true for another pathway $\Psi_{k'}$. When Ψ_k and $\Psi_{k'}$ interact by the operations of transduction, different components are deleted due to the fact that they cannot be sustained. Then we get the final result. The advantage of this scheme is that the number of candidates has no relation to the molecules we set in advance. The recursive generation of pathways is executed to sift out the less suitable candidates. In the meantime, The truth values according to our criterion, the common reactants in pathway Ψ_k (k=0, 1, ...) ensure that the related pathways are sustained. This continues to loop until rewriting stops at the final stage, i.e., the "reactions" concerned do not produce any more new "reactants". The term "reaction" refers to the interaction of pathways under the context of reactive system theory. The term "reactant" refers to the factors used to implement the reaction processes.

At this point, we need to check whether the solution has been obtained according to the following rule:

> If the "reactions" concerned do produce any more new "reactants", the computing process goes to the next step.
> If the "reactions" concerned do not produce any more new "reactants", we must update the population and let the computing process go to Step 1.

Step 2: Judging by the terminal criterion.

The criterion Ξ_H to judge the halting of the entire process is:

> whether the final variable form of the candidates is confirmed as the solution when no more new pathways emerge.

At this point, the existing pathways are identical in sets of pathways. Finally, after the result is confirmed, the final solution will be decided as the output.

4. Conclusion

In this abstract, we have briefly reported our result of studying (hyper)graph rewriting embedded by algebriac operators. The next step of our work will be systematical analysis of the quantitative relationship between the neighborhood selection and the complexity derived from the corresponding operators.

Acknowledgment

J.-Q.Liu sincerely thanks Prof. Masami Ito for his teaching and academic advices, Prof. Tatsuhiko Saito, Prof. Teruo Imaoka, Prof. Yuji Kobayashi, Prof. Hidenosuke Nishio, Prof. Katsumasa Watanabe and Prof. Reiji Nakajima for their suggestions on theoretical computer science.

Main References

[1] B. Courcelle, The expression of graph properties and graph transformations in monadic second-order logic, Chapter 5 of the "Handbook of graph grammars and computing by graph transformations, Vol. 1: Foundations", G. Rozenberg (ed.), World Scientific (New-Jersey, London), 1997, pp. 313-400.

[2] Robin Milner, Bigraphical reactive systems: basic theory, September 2001 (http://www.cl.cam.ac.uk/users/rm135/bigraphs.ps.gz).

[3] Jian-Qin Liu and Katsunori Shimohara, Graph rewriting in topology I: Operators and the Grammar, SIG-FAI-A101-4(9/21), Sept.21, 2001, pp.21-26.

[4] Jian-Qin Liu and Katsunori Shimohara, Graph rewriting in topology II: computability, Proceedings of the 5th symposium on algebra, language and computation (edited by T. Imaoka and K. Shoji, Shimane Univ., Nov.17-18, 2001), pp. 63-68.

[5] Teruo Imaoka, Preface of RIMS Kokyuroku 1222 "Algebraic Semigroups, Formal Languages and Computation" (July 2001, Research Institute for Mathematics Sciences, Kyoto University, Kyoto, Japan), Edited by Teruo Imaoka.

[6] Yuji Kobayashi, Masashi Katsura, Linear Markov properties and undecidability, RIMS Kokyuroku 1222 "Algebraic Semigroups, Formal Languages and Computation" (July 2001, Research Institute for Mathematics Sciences, Kyoto University, Kyoto, Japan), Edited by Teruo Imaoka, pp.98-104.

[7] Isamu Inata and Yuji Kobayashi, Monoids with subgroups of finite index and braid inverse monoid, Proceedings of the 5th symposium on algebra, language and computation (edited by T. Imaoka and K. Shoji, Shimane Univ., Nov.17-18, 2001), pp. 24-28.

[8] Jian-Qin Liu and Katsunori Shimohara, Graph rewriting in topology III: computation embedded with Heegaard splittings, IPSJ SIG Notes, vol.2002, no.7, Jan.24, 2002, p.43.